KubeCon | CloudNativeCon

North America 2017

Regain control thanks to Prometheus

Guillaume Lefevre, DevOps Engineer, *OCTO Technology*
Etienne Coutaud, DevOps Engineer, *OCTO Technology*

About us

Guillaume Lefevre
DevOps Engineer, *OCTO Technology*
@guillaumelfv

Etienne Coutaud
DevOps Engineer, *OCTO Technology*
@etiennecoutaud

What we are dealing with

- **Track** every events of a **package lifecycle**
- Aggregate and compute events to **generate** relevant **business data**
- **Store and serve** results for consultation or further computing

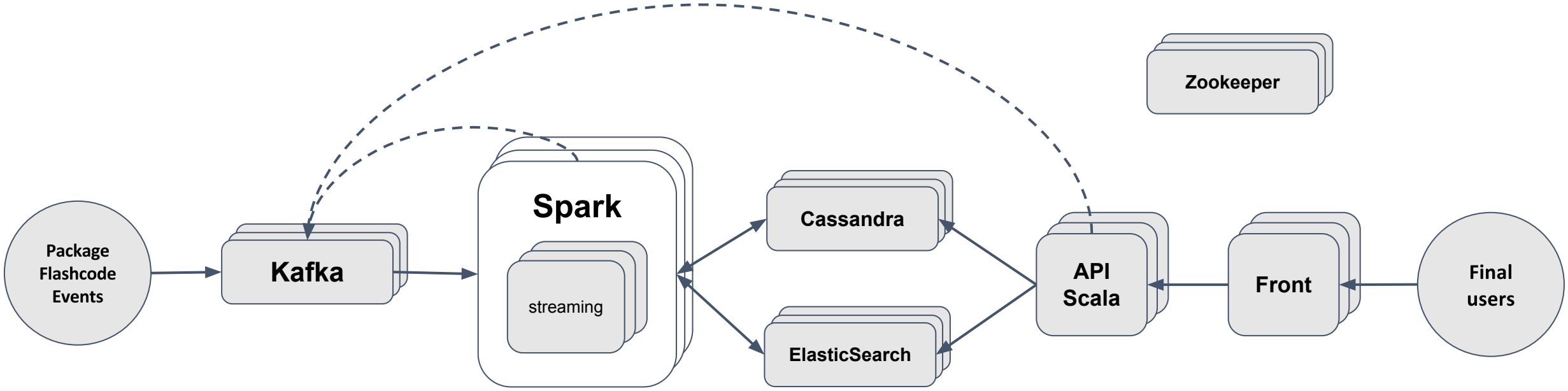
What we are dealing with

2,000,000
packages / day

20,000,000
events / day

200 processes
running 24/7

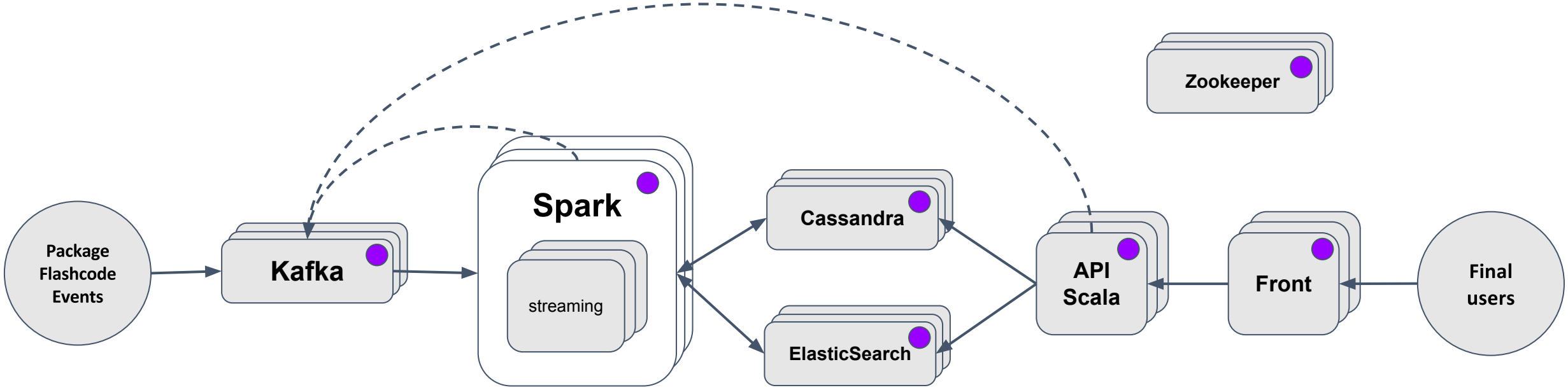
CQRS Architecture



- ~ 90 servers
- Whole configuration managed with

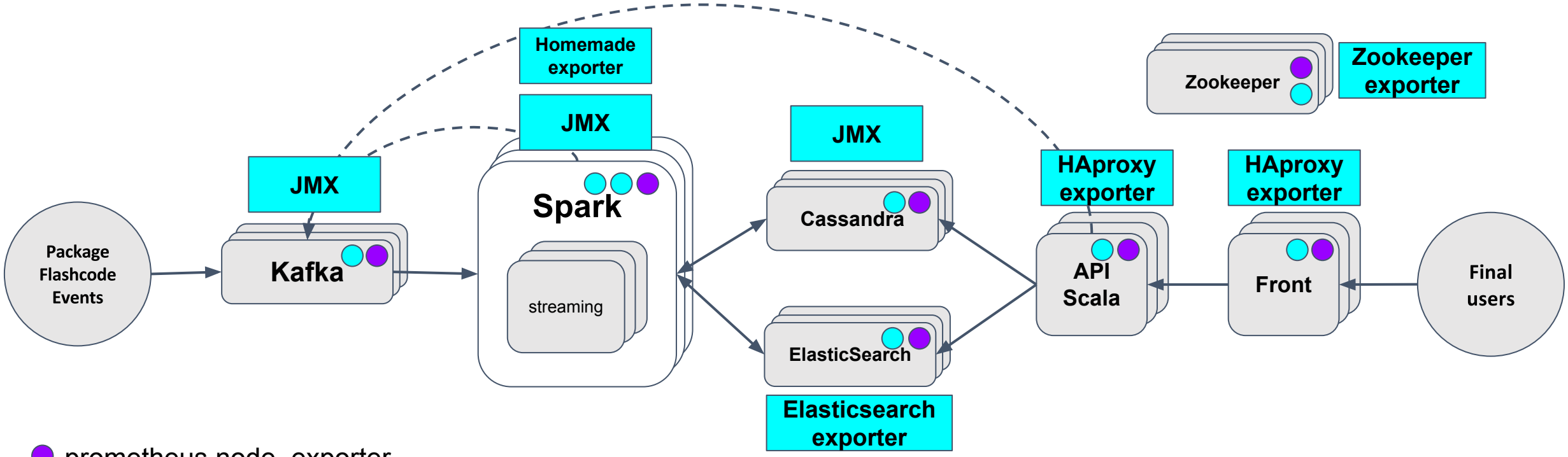
ANSIBLE

Monitoring system metrics



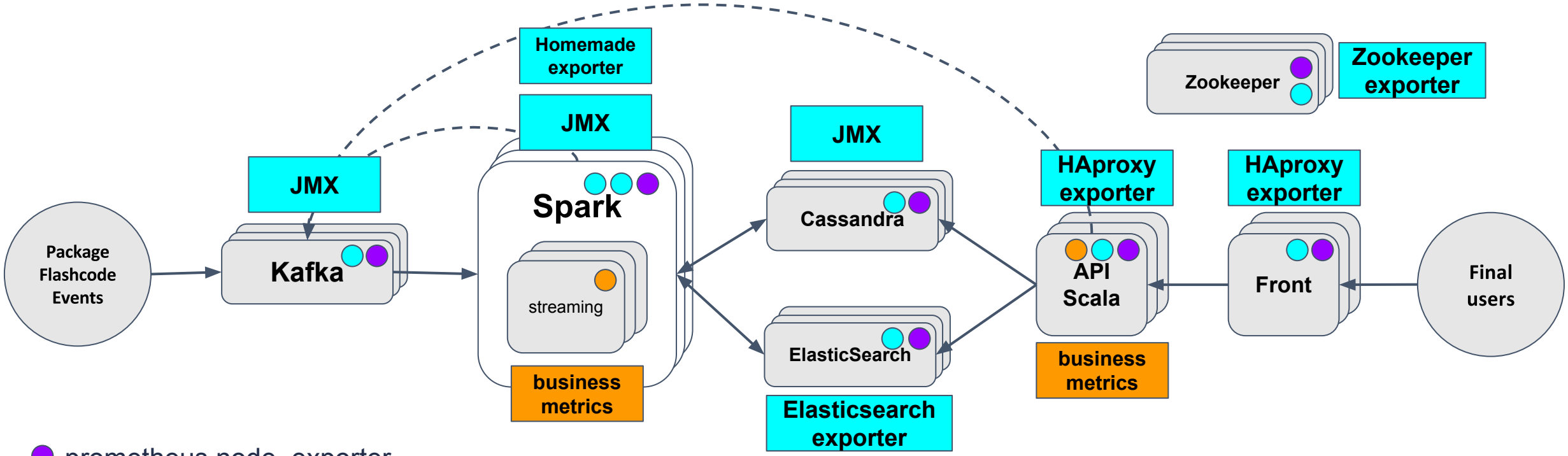
● prometheus node_exporter

Monitoring middleware metrics



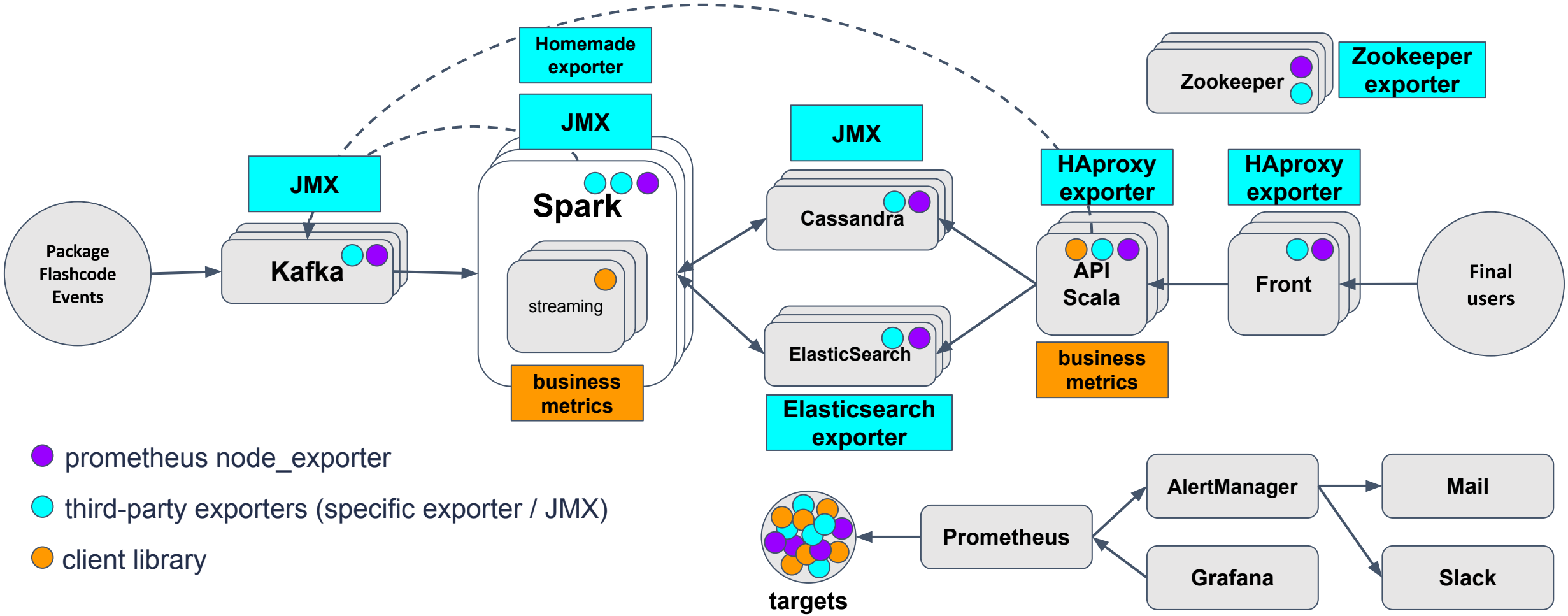
- prometheus node_exporter
- third-party exporters (specific exporter / JMX)

Monitoring business metrics



- prometheus node_exporter
- third-party exporters (specific exporter / JMX)
- client library

Full stack


Homemade exporter
JMX
JMX
JMX
Zookeeper
Zookeeper exporter
HAproxy exporter
HAproxy exporter
Package Flashcode Events
Kafka
Spark
streaming
business metrics
Cassandra
ElasticSearch
Elasticsearch exporter
API Scala
business metrics
Front
Final users
prometheus node_exporter
third-party exporters (specific exporter / JMX)
client library
targets
Prometheus
AlertManager
Grafana
Mail
Slack

Data volume

~ 250
endpoints scrapped

~ 15,000
different metrics

~ 50 GB
data / day

Enabling metrics: node_exporter

Wrapped with *systemd*

```
$ ./node_exporter --web.listen-address 0.0.0.0:9100 --collector.systemd
```

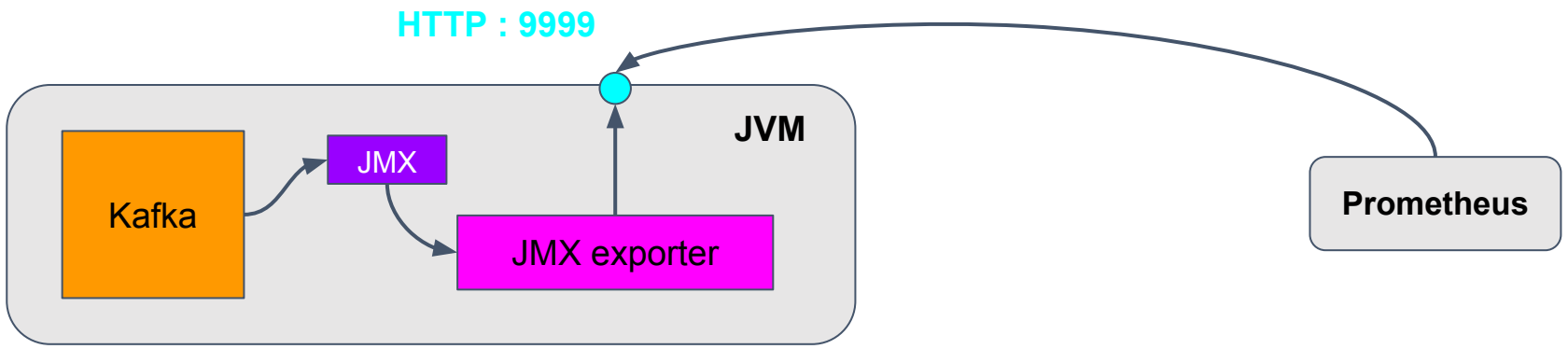
Have a look at the "***disabled by default***" **collectors**

https://github.com/prometheus/node_exporter

Enabling metrics: JMX

Use the exporter as a java-agent at JVM startup

```
KAFKA_OPTS="-javaagent:/var/lib/jmx-exporter/jmx_prometheus_javaagent.jar=9999:/var/lib/jmx-exporter/kafka.yml"
```



Enabling metrics: Scala app

Use the scala library to define metrics

```
object SparkAppMetrics extends ContextLogger {
  private lazy val handledXMLCounter: SparkCounter = ReacUserMetricsSystem.counter("handledXMLCounter")
  private lazy val measuredXMLCounter: SparkCounter = ReacUserMetricsSystem.counter("measuredXMLCounter")
  private lazy val handledMessagesCounter: SparkCounter = ReacUserMetricsSystem.counter("handledMessagesCounter")
  private lazy val rejectedMessagesCounter: SparkCounter = ReacUserMetricsSystem.counter("rejectedMessagesCounter")
  private lazy val succeededMessagesCounter: SparkCounter = ReacUserMetricsSystem.counter("succeededMessagesCounter")
```

Inject metrics into Spark context

```
object InjectorReacApp extends SparkApp ({ (ssc: StreamingContext) =>

  ReacUserMetricsSystem.load("InjectorReacMetrics")(ssc.sparkContext)
```

SelfHealing using AlertManager + Jenkins

```
ALERT SparkWorkerDataFull
  IF node_filesystem_free{job='spark_system_metrics} < 0.05
  FOR 5m
  LABELS {
    severity = "high",
    environment = "production",
    selfhealing = "true",
    jenkins_job = "clean_data_worker"
  }
  ANNOTATIONS {
    summary = "Disk /data on {{ $labels.instance }} 95% full"
  }
```

AlertRule in Prometheus server

AlertManager config file

```
route:
  - match:
      selfhealing: true
      jenkins_job: clean_data_worker
    receiver: jenkins-selfhealing-clean_data_worker

receivers:
- name: jenkins-selfhealing-clean_data_worker
  webhook_configs:
  - url: "http://jenkins:8080/job/production/clean_data_worker/build"
```

Our favorite Prometheus queries

- *Topk*: used to get largest K elements by sample value, useful to extract components from the whole metrics

Example: Get the 3 servers with the least free RAM percentage

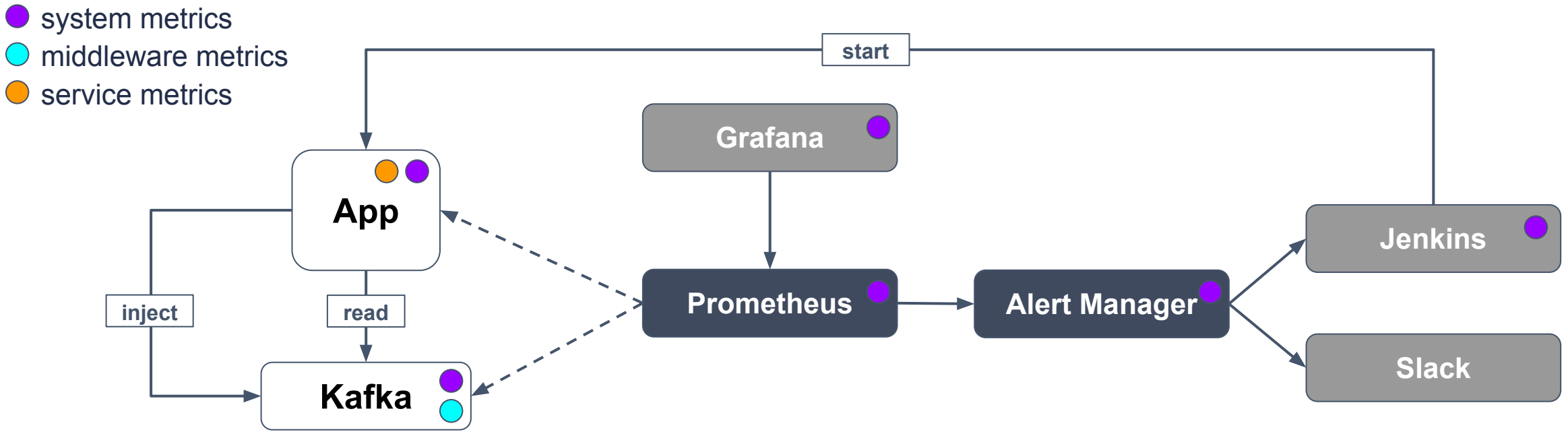
```
topk(3,(1 - node_memory_MemFree / node_memory_MemTotal))
```

- *predict_linear*: used to predict a value with a simple linear regression, useful to extrapolate disk filling

Example: Predict filesystem fullness in less than 6h

```
predict_linear(node_filesystem_free{mountpoint='/data'}[1h], 6 * 3600) < 0
```

Demo time



Code available at: https://github.com/EtienneCoutaud/kubecon2017-demo.git

How do ops use it?

- Self-healing & Alerting help you focus on **delivering more value**
- Predict failure and correct it to **prevent outage** and business loss
- **Faster troubleshooting** and root cause pinpointing
- **Tune and improve configuration** middleware parameters
- Discover bottleneck and **improve** overall platform **performance**

How do dev use it?

- Easier performance **testing** and **benchmarking**
- Bring **visibility** and confidence into the platform
- Give a comprehensible **view** of the platform & **KPIs** to top management
- Allow **developers** to add **their own metrics, dashboards and alerts** at application level

Tip: Never stop tweaking

Continuous improvement is the best way to:

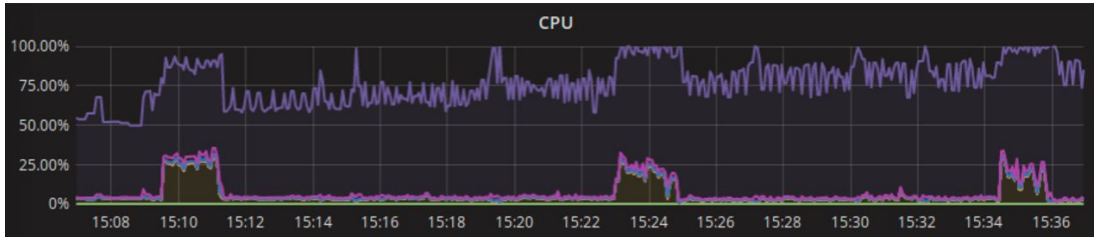
- Choose the **right metrics**
- Choose **accurate thresholds**
- **Prevent** alert **fatigue**

Tip: Mind your NTP

- Ensure all your applications, servers and monitoring platform are **in-sync**
- Servers and browser **clocks must not be too far apart**

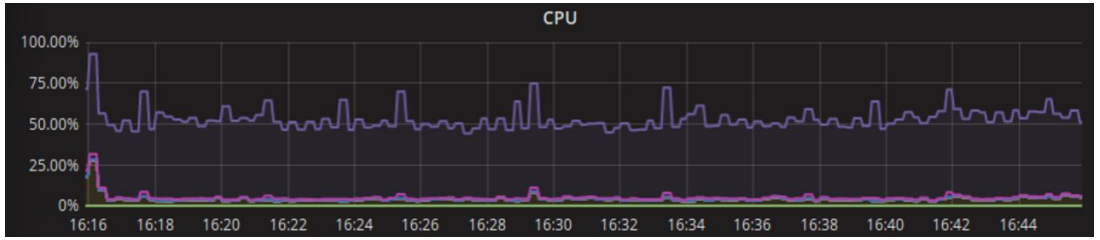
Trick: Scraping interval tradeoff

low scraping interval (5s)



- Almost real-time
- CPU intensive (increase with endpoints)
- Robust Prom server or federation needed

higher scraping interval (15s)



- Metrics “lag effect”
- Use less server resources
- Sustain more endpoints

Choose wisely based on your business requirements!

Golden tip

Work with your users!

What's next?

- Upgrade to 2.0 to improve performance
- Explore advanced features (federation, HA & third party-storage)
- Improve trigger precision and alarms
- Set up dashboard drilling for faster investigation
- Automate incident responses because we are lazy
- Monitor the underlying infrastructure

Take-away

- **FullStack Monitoring**
- Fast to deploy, **long to master**
- Share it with **everybody** / Provide **feedback** to everyone
- **Changes the way of working**
- Prometheus has **many** differents **exporters**, you will surely find yours!
- The format makes it **easy** to build a custom exporter